LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-404711

A Newton-Krylov Solver for Implicit Solution of Hydrodynamics in Core Collapse Supernovae

D. R. Reynolds, F. D. Swesty, C. S. Woodward

June 16, 2008

SciDAC 2008
Seattle, WA, United States
July 13, 2008 through July 17, 2008

Disclaimer

A Newton-Krylov Solver for Implicit Solution of Hydrodynamics in Core Collapse Supernovae

Daniel R. Reynolds[1] **and F. Douglas Swesty**[2] **and Carol S. Woodward**[3, 4]

[1] Department of Mathematics, University of California at San Diego, 9500 Gilman Dr., La Jolla, CA 92093-0112, USA
[2] Department of Physics and Astronomy, SUNY at Stony Brook, Stony Brook, NY 11794-3800, USA
[3] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, USA

E-mail: drreynolds@ucsd.edu, dswesty@mail.astro.sunysb.edu, cswoodward@llnl.gov

Abstract. This paper describes an implicit approach and nonlinear solver for solution of radiation-hydrodynamic problems in the context of supernovae and proto-neutron star cooling. The robust approach applies Newton-Krylov methods and overcomes the difficulties of discontinuous limiters in the discretized equations and scaling of the equations over wide ranges of physical behavior. We discuss these difficulties, our approach for overcoming them, and numerical results demonstrating accuracy and efficiency of the method.

1. Introduction

Due to a large disparity in time scales, a multitude of relevant physical processes, and solution components spanning a wide range of scales, modeling the collapse of a massive stellar core to produce either a hot proto-neutron star or a supernova explosion proves to be an inherently challenging numerical problem. The model must describe the hydrodynamic behavior of hot, dense matter with complex composition and must describe the flow of radiation, in the form of neutrinos, through the matter. During the cooling phase of these processes, neutrinos carry off most (approximately 99%) of the gravitational potential energy released by the collapse of the star's core, with the remaining 1% or so of the energy contributing to the observed supernova explosion. This general scenario was confirmed through detection of neutrinos by two independent experiments from supernova SN1987A [1, 2]. Simulations of such phenomena must account for both the explosion and cooling phases, covering a time period of tens of seconds [3]. Thus algorithms for these simulations must allow high accuracy over long periods of time.

In such models a set of combined fluid dynamic and radiation transport equations must be solved to describe the evolution of the system from its initial state. If explicit time integration methods are applied for the hydrodynamic processes, time steps will be limited by the CFL stability constraint, typically requiring steps less than $1\mu s$ [4]. Furthermore, coupling between matter and radiation occurs on a time scale even faster than the CFL limit, resulting in significant

[4] Work of this author was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

stiffness in the problem. To date, no radiation-hydrodynamic simulation for evolution of a collapsed core over the period of tens of seconds has been possible using explicit methods. Explicit calculations [5, 6] have been limited to short ($< 1s$) periods of time. Long timescale cooling calculations [3, 7] have had to assume the collapsed core is in hydrostatic equilibrium. Since implicit methods allow high-order-accurate solutions, can be robust for stiff problems, and are not limited by CFL stability constraints, their use is thus highly desirable for these models.

The use of implicit methods introduces its own set of challenges beyond those strictly pertaining to the relevant physics. First, the governing equations are nonlinear and sparse, suggesting the use of Newton-Krylov (hereafter NK) iterative methods [8]. Second, discretizations employed to capture the behavior of hydrodynamic shocks present in the system introduce limiters that result in discontinuities in the nonlinear system. These discontinuities are problematic from the standpoint of the convergence theory of NK methods. A third difficulty with using implicit methods arises from the fact that the model equations describe the evolution of matter and radiation over a range of densities and physical length scales spanning many orders of magnitude from the center of the collapsing core to the outer edges of the computational domain. This wide range of values means that the discretized equations involve scaling issues that can present difficulties for NK methods.

Another element of complexity arises due to the size of the discrete systems. Radiation transport models of core collapse supernovae must describe the evolution of the distribution of neutrinos across a large number of frequencies, thus adding a spectral dimension to the space-time discretization, and increasing the size of the resulting nonlinear and linear systems that must be solved. In multiple spatial dimensions, this problem then requires the use of massively parallel computing resources in order for the memory and computation requirements to even be tractable. Thus, the approach required for solution of the discrete systems must be scalable to large parallel computing platforms.

In this work, our goal has been to develop a nonlinear solver framework employing Newton-Krylov methods robustly on a variety of radiation-hydrodynamic discretization schemes. NK methods are typically used for large, sparse problems through combining Newton's method with a scalable preconditioned Krylov method for solving the linear systems within each nonlinear iteration [8]. We targeted NK methods for these problems due to the fast, quadratic convergence properties of Newton's method. NK methods have been very effective in solving large, nonlinear systems in a number of application areas; for details we refer the reader to the review article [9]. In order to effectively test our NK methodology on the applications of interest, we consider the problem of cooling the collapsed core in spherical symmetry (thus creating a 2-D nonlinear system in space and neutrino frequency).

This paper is organized as follows. In the next section, we present the implicit fluid dynamics models we use. The third section describes our NK method, and the fourth section discusses some software packages that provide efficient and robust implementations of NK methods. The fifth section presents numerical results, and the last section gives some conclusions.

2. Implicit Hydrodynamics Models

While our ultimate goal is to use NK methods to solve the full set of radiation hydrodynamic equations in an implicitly-coupled fashion, in this paper we consider only the most difficult portion of the system from a nonlinear solver viewpoint, the nonlinear hydrodynamic equations. We handle the nonlinear radiation transport equations through an operator-split approximation. The use of NK methods to treat the neutrino radiation has been described elsewhere [10], and future work will investigate a full nonlinear coupling of these two implicit systems. Our focus in this work is to establish a robust methodology to effectively deal with the challenges of implicit hydrodynamics in the context of proto-neutron star and supernova simulations. The implicit approach we consider addresses both Lagrangean and Eulerian formulations of compressible

hydrodynamics with differing spatial meshes and discretization schemes.

We first consider the Langragean formulation which is applicable in the initial stellar collapse phase of a supernova. In this phase, a comoving mass coordinate m follows the collapse over many orders of magnitude in the radial coordinate r (which evolves in time). The Lagrangean formulation for compressible hydrodynamics is given by [11]

$$\frac{Dv}{Dt} = \frac{Gm}{r^2} - 4\pi r^2 \left(\frac{\partial(P_m + P_R)}{\partial m}\right) \equiv F^{lv}, \tag{1}$$

$$\frac{Dr}{Dt} = v \equiv F^{lr}, \tag{2}$$

$$\frac{D}{Dt}\left(\frac{E_m}{\rho}\right) = -P_m \frac{D}{Dt}\left(\frac{1}{\rho}\right) \equiv F^{le}, \quad \text{and} \tag{3}$$

$$0 = \frac{\partial m}{\partial r} - 4\pi r^2 \rho \equiv G^{lm}, \tag{4}$$

corresponding to conservation of momentum, the definition of velocity, conservation of energy, and self-gravity. In (1)-(4), v is the matter velocity, ρ is the matter density, E_m is the matter energy density, and P_m is the matter pressure. The radiation pressure P_R is considered an input from the radiation solver in a standard operator-split approach. The operator D/Dt is used to denote the Lagrangean (comoving frame) time derivative. The system of equations is closed through an equation of state of the form

$$E_m = E_m(T, \rho) \quad \text{and} \quad P_m = P_m(T, \rho), \tag{5}$$

in terms of the matter temperature T. The equations are spatially discretized using the Richtmyer-Morton finite-difference method using a staggered grid and an artificial viscosity scheme to treat shocks (see [11], p. 485). The discrete analogs of (1)-(4) for each cell form the coupled set of sparse, nonlinear equations that must be solved to advance each time step, as described in Section 3.1. The system is defined by the vector of unknowns and the vector of nonlinear equations. In this Lagrangean discretization the vector of unknowns is given by $x^T = \left(\rho_{i+1/2}, T_{i+1/2}, r_i, v_i\right), i = 1, \ldots, N$, where integer indices denote quantities defined at cell edges, half-integer indices denote quantities defined at cell centers, and N denotes the total number of active cells. The vector of nonlinear equations corresponds to the discretized versions of equations (1)-(4) within each cell. Let us denote this system of unknowns and equations as System I, and note that the total size of the system is $4N$.

In the initial phase of stellar collapse it is desirable to employ a Lagrangean hydrodynamic scheme as the core of the star contracts from a radius of several thousand kilometers to just a few hundred kilometers [4]. The Lagrangean scheme allows coordinates to comove with the collapsing core. After collapse it is typically more desirable to consider an Eulerian formulation where the coordinates are fixed in space. We consider two common Eulerian formulations, differing in the choice of energy equation. One formulation employs a gas energy equation and the other a total energy equation. Each of these approaches has merits and limitations. The former approach ensures that the First Law of Thermodynamics is satisfied while the latter ensures the conservation of total energy. In spherical symmetry the former approach is given by

$$\frac{\partial \rho}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho v\right) \equiv F^{ec}, \tag{6}$$

$$\frac{\partial E_m}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 E_m v\right) - P_m \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 v\right) \equiv F^{eg}, \quad \text{and} \tag{7}$$

$$\frac{\partial (\rho v)}{\partial t} = -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \rho v^2\right) - \left(\frac{\partial (P_m + P_r)}{\partial r}\right) - \frac{Gm\rho}{r^2} \equiv F^{ev}, \tag{8}$$

corresponding to conservation of mass, evolution of gas energy, and conservation of momentum. These are augmented with the self-gravity equation (4), which remains unchanged from the Lagrangean formulation.

Such a formulation has been used in the explicit ZEUS-2D code [12]. We consider the implicit generalization of this scheme using a staggered mesh, an artificial viscosity, and a van Leer monotonic limiter to treat shocks (see [12] for details). In this discretization the vector of unknowns is given by $x^T = \left(\rho_{i+1/2}, T_{i+1/2}, v_i, m_i\right), i = 1, \ldots, N$, where the mass coordinate m evolves over time and the spatial coordinate r is held fixed. The vector of nonlinear equations defining this system corresponds to the discretized versions of (6)-(8) and (4) within each cell. We denote this as System II, and note that the total size of the system is again $4N$.

An alternative formulation replaces (7) with an equation for conservation of total energy,

$$\frac{\partial(E_m + \frac{1}{2}\rho v^2)}{\partial t} \;=\; -\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\left((E_m + \frac{1}{2}v^2 + P_m)v\right)\right) \;\equiv\; F^{et}. \tag{9}$$

This approach is used in many finite-volume fluid dynamics methods. We consider the implicit extension of the widely used LCPFCT discretization scheme [13], which is an extension of the pioneering FCT scheme [14]. This approach uses a cell-centered finite-volume mesh with limiters to maintain monotonicity and capture shocks. In this discretization the vector of unknowns is given by $x^T = \left(\rho_{i+1/2}, T_{i+1/2}, v_{i+1/2}, m_{i+1/2}\right), i = 1, \ldots, N$, and the vector of equations is given by the discretized versions of (6), (8), (9) and (4) within each cell. We denote this as System III, and note the size of the system is also $4N$.

3. Integration and Solution Approach

This section overviews the implicit time integration approach, along with the fundamental components of our NK method.

3.1. Temporal Integration with θ-scheme

We employ a two level θ-scheme for time integration. This formulation allows switching between explicit and both first and second order implicit integration methods based on the continuous parameter θ. For the relevant spatially I, II, or III, this scheme is

$$\frac{V(x^n) - V(x^{n-1})}{\Delta t} \;=\; \theta F(x^n) + (1-\theta)F(x^{n-1}), \tag{10}$$

$$G(x^n) \;=\; 0, \tag{11}$$

where $F(x^n)$ represents the set of discretized time-dependent equations {(1)-(3)}, {(6)-(8)}, or {(6), (8), (9)}, evaluated at time t^n, and $G(x^n)$ represents the discretized constraint (4). We write $V(x)$ to denote that the time-derivatives include functional forms of the unknowns, and the form of V is readily apparent through inspection. When $\theta = 0$ this scheme reduces to the forward Euler explicit method. When $\theta = 1$ the scheme is the first order backward Euler implicit method. When $\theta = 0.5$ the scheme is the second order Crank-Nicholson implicit method.

3.2. Newton-Krylov Nonlinear System Solver

When advancing (10)-(11) with $\theta > 0$, a nonlinear system of equations given by,

$$H\left(x(t^n)\right) = 0, \tag{12}$$

must be solved. Here $x(t^n) \in I\!R^{4N}$ is the vector of unknowns approximated at time t^n and H is the vector comprised of (10) and (11).

The Newton-Krylov algorithm was first developed for use in solving discrete nonlinear systems arising from PDEs in [15]. This method leverages the quadratic convergence of Newton's method with the robustness and scalability of preconditioned Krylov methods. Details of the algorithm can be found in [9] and [8]. The basic algorithm for solution of the system $H(x) = 0$ is:

Start with an initial condition x^0 at time t^0:
For each time step $n = 1, \ldots, N$, find $x^n \approx x(t^n)$ by Newton's method:

1. *Start with an initial Newton iterate $x^{n(0)}$ (typically $x^{n(0)} = x^{n-1}$)*
2. *For each Newton iteration $j = 1, 2, \ldots$*
   a. *Using a Krylov method, approximately solve for s^j,*
   $J_H\left(x^{n(j)}\right) s^j = -H\left(x^{n(j)}\right)$ *such that* $\left\| J_H\left(x^{n(j)}\right) s^j + H\left(x^{n(j)}\right) \right\| \leq ltol.$
   Within each Krylov iteration perform:
      i. One matrix-vector multiply with $J_H\left(x^{n(j)}\right)$
      ii. One preconditioner solve
   b. *Update the Newton iterate, $x^{n(j+1)} = x^{n(j)} + \lambda s^j$*
   c. *Test for convergence to stop the iteration, $\left\| H\left(x^{n(j+1)}\right) \right\| < htol.$*

Here, $J_H\left(x^{n(j)}\right) \in \mathbb{R}^{4N \times 4N}$ is the Jacobian of the nonlinear function $H(x)$ evaluated at the previous Newton iterate $x^{n(j)}$, and $\lambda \in (0, 1]$ is a line search parameter chosen to help globalize the method [8]. Information on Krylov methods and their use with Newton's method can be found in [8, 16]. The Krylov methods most commonly used within an NK framework are GMRES [17] and BiCGStab [18] as these are general to non-symmetric systems as is typical in multi-physics applications.

As noted in the algorithm description above, Krylov methods require a matrix-vector multiply at each iteration. In fact, there is no other requirement or use of the linear system matrix. Since this matrix arises from the nonlinear system, instead of constructing $J_H(x)$ directly we can approximate the action of the Jacobian on a vector with a directional difference,

$$J_H(x)w \approx H(x + \sigma w) - H(x)/\sigma, \tag{13}$$

while still preserving convergence of the overall Newton method [19]. Thus, as long as the nonlinear function can be evaluated at each linear iteration, the Newton-Krylov solve can proceed without analytically calculating Jacobian matrices.

Although very robust, Krylov methods can often slow or stall in their convergence. As a result, we apply a right preconditioner, P, within Step 2.a. via the transformation

$$J_H s = -H \quad \longrightarrow \quad (J_H P^{-1})(Ps) = -H. \tag{14}$$

3.3. Solver Customizations for Astrophysical Hydrodynamics

While the standard NK algorithm works in theory for sufficiently-differentiable nonlinear problems, astrophysical hydrodynamics presents a number of challenges to these methods. We describe the key ingredients required for application of NK algorithms to our systems.

As described in Section 2, flux limiters and artificial viscosity terms are included in the models with the goal of increasing the stability of numerical methods in the presence of shocks. Unfortunately, these terms are typically enabled through non-differentiable "if" statements in a code. As such, the nonlinear functional $H(x)$ is not in general Lipschitz continuous, as is required for Newton convergence theory, and the equations themselves may even change between one evaluation of H and the next. In practice, such effects may result in non-convergence of a matrix-free Krylov linear solver, since the multiple evaluations of H in (13) may be based on different

equations, with possibly differing artificial viscosities or flux terms. To ameliorate this difficulty, we "freeze" the choice of these algorithmic nonlinearities throughout each Newton iteration. The resulting Krylov iteration converges without difficulty, leaving any differentiability issues to the more forgiving globalized Newton solver, which may require at most a few additional iterations to resolve the solution. We accomplish this "freeze" through approximating the Jacobian matrix $J_H(x)$ directly, again using finite-differences, but now to compute the specific Jacobian entries,

$$\left[J_H\left(x\right)\right]_{i,j} \approx H_i\left(x + \sigma e_j\right) - H_i\left(x\right)/\sigma, \quad i,j = 1,\ldots,4N. \tag{15}$$

where e_j corresponds to the unit vector formed with zeros in all but the jth entry, and $H_i(x)$ corresponds to the ith entry in the nonlinear residual vector. In addition, since the hydrodynamics equations involve a fixed stencil size, the large majority of these entries are zero. We exploit this feature through the use of *colored finite-difference* methods to perform the Jacobian approximation (15) using only a small fixed number of evaluations of H. Through approximating these entries, we effectively "smooth" out discontinuities present in the true Jacobian enabling nonlinear convergence. In addition, with these precomputed Jacobian entries, we easily construct a block-diagonal preconditioning matrix P through extracting only the spatially-local entries that couple the variables together within a given cell.

Typically, Newton methods are declared "converged" when a norm of the nonlinear residual is below a given tolerance, i.e. $\|H(x)\| < htol$. However, for coupled systems, each equation may be expressed in a different scale (or units). As a result, even when a Newton iteration x is far away from the solution, some equations may have very "small" residuals compared with others. Hence, the choice of convergence criteria becomes very important, in that it must measure convergence of *all* equations, irrespective of whether the units result in large or small residuals. We employ a scaled root-mean-squared norm in measuring convergence,

$$\|H(x)\| = \left(\frac{1}{4N}\left\|DH(x)\right\|_2^2\right)^{1/2}, \tag{16}$$

where $D = \mathrm{diag}\left(d_1^{-1}, d_2^{-1}, \ldots, d_{4N}^{-1}\right)$, with d_i corresponding to the typical magnitude of the given equation at that spatial location, effectively re-normalizing all of the equations to remove unit discrepancies. To this end, we choose these scaling values as

$$d_i = \begin{cases} 1, & \text{Equations (1), (6), and (9)} \\ r^{n-1}, & \text{Equation (2)} \\ E_m^{n-1}\rho^{n-1}, & \text{Equation (3)} \\ m, & \text{Equation (4), System I} \\ \rho^{n-1}, & \text{Equation (4), Systems II and III} \\ E_m^{n-1}, & \text{Equation (7)} \\ p^{n-1}/\rho^{n-1}, & \text{Equation (8)(sound speed)} \end{cases}. \tag{17}$$

Finally, unlike standard nonlinear systems of equations, the systems I, II and III involve variables with implicit positivity constraints, i.e. density and energy must be strictly positive. For astrophysical problems, the values of these variables can vary by orders of magnitude throughout the spatial domain. As a result, small errors due to inexact solver tolerances or even floating-point precision can result in violations of constraints when solution values get small. However, the standard Newton-Krylov algorithm does not account for constraints on the variables. We employ a novel transformation to logarithmically rescale these sensitive variables for use within the Newton solver. For example, instead of using values for the density ρ, we use $\tilde{\rho} = \log(\rho)$ in the Newton solver and unknown vector x. This transformation adjusts the nonlinear functional so that it depends on the nicely scaled $\tilde{\rho}$ instead of ρ. The Newton updates

are performed on the new unconstrained variable, $\tilde{\rho}^{n(j+1)} = \tilde{\rho}^{n(j)} + \lambda s^{j+1}$, which cannot result in constraint violations in the density $\rho = \exp(\tilde{\rho})$. We use such scalings in System I on the variables $\{\rho, T, r\}$, and in Systems II and III on the variables $\{\rho, T, m\}$. This type of transformation increases the nonlinearity of the functional $H(x)$ on the solver arguments (e.g. $\tilde{\rho}$), which in practice results in approximately one additional Newton iteration per step.

4. Software

Our implementation of the nonlinear solver portion of the simulation code uses the SciDAC TOPS solver package, KINSOL from the SUNDIALS suite [20, 21]. SUNDIALS is a C language package written with the goal of easily being included into existing physics codes. The implementation of Newton-Krylov methods within KINSOL is data structure neutral, in that the implementation does not depend on exactly what types of data structures the user code has for variables and equations. We note that robust implementations of NK methods are also available in the TOPS C++ package, Trilinos [22], and the TOPS C package, PETSc [23]. The physics routines in our code, RH1D, are written in F95 to achieve good performance. The core of the physics is contained in routines to evaluate the nonlinear equations. The code to accomplish the colored finite-difference approximation of the Jacobian is also written in F95 to obtain peak efficiency as are all vector operations needed to implement the vector operations needed by SUNDIALS.

5. Numerical Results

In this section we show results of applying the implicit solution method to two analytic solution problems and a proto-neutron star cooling problem.

5.1. Analytic Solution Testing

The analytic tests we performed consisted of shock tube problems, blast wave problems, and hydrostatic equilibrium problems. We lack space to recount these tests in full detail here, but present highlights of the shock tube and blast wave. Our first reference problem is the Sod shock tube [24]. The numerical performance of the three hydrodynamic discretizations, described in Section 2, on this problem are shown in Figures 1-3, along with the analytical reference solutions (solid lines). In these tests the time step was set by choosing a CFL factor of 0.5 and using

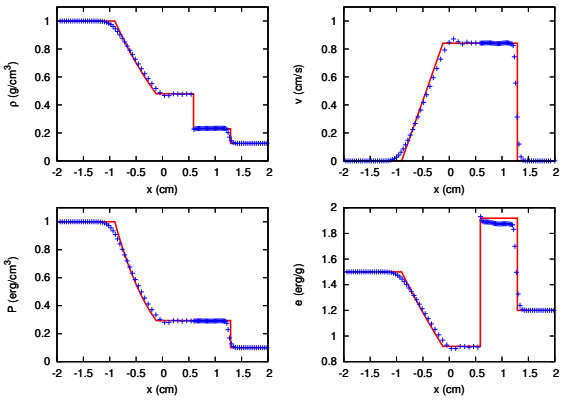

Figure 1. The Sod shock tube for the implicit Lagrangean scheme, I.

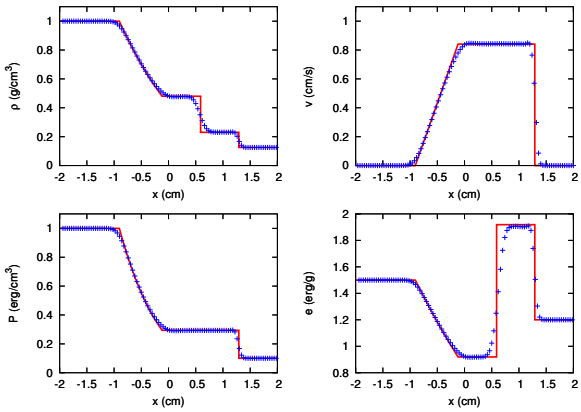

Figure 2. The Sod shock tube test problem for the implicit staggered mesh scheme, II.

$\theta = 1$ for the Lagrangean scheme and $\theta = 0.55$ for the Eulerian schemes. The computational domain was discretized into $N = 100$ zones. The full details of the problem setup are described

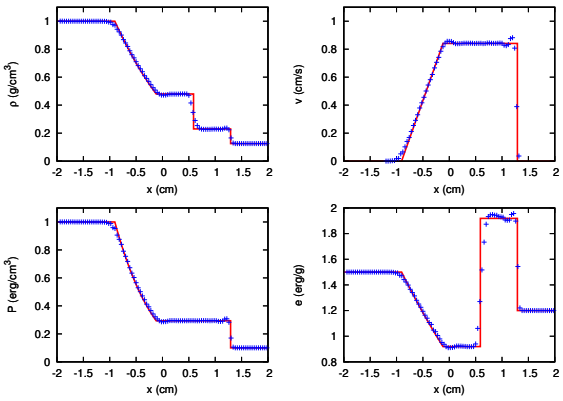

Figure 3. The Sod shock tube test problem for the implicit FCT scheme, III.

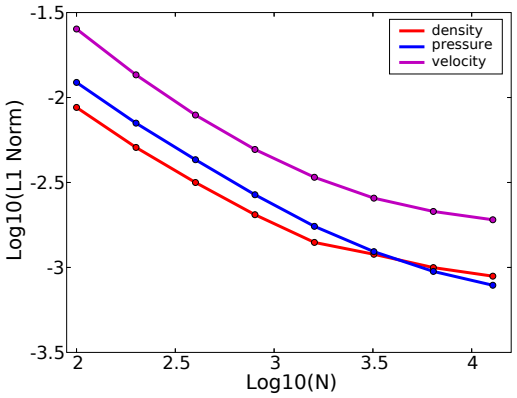

Figure 4. Convergence, of the implicit Lagrangean scheme for the Sod shock tube.

in [25, 10]. Our tests indicate that the implicit algorithms yield comparable numerical results to each other, and to comparable explicit algorithms (see [13, 12, 25] for comparison). Increasing resolution in these tests (Fig. 4) shows that the methods are converging as expected. We also

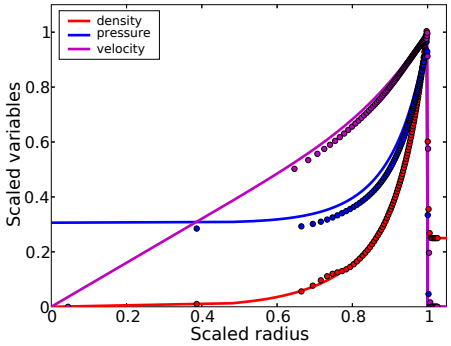

Figure 5. The analytic and implicit Lagrangean numerical solutions of the Sedov-Taylor problem. Radius is scaled by the analytic shock radius, and variables are scaled relative to the values at the shock front.

present results from tests on the Sedov-Taylor blast wave problem. Results for the implicit Lagrangean algorithm are shown in Fig. 5. This problem is particularly challenging to the nonlinear solver because the blast-wave test creates an extremely dynamic set of conditions. The initial conditions for this test are described in [10] and are chosen to create a problem where the range of variables across the computational domain varies by eight orders of magnitude. The ensuing blast wave sweeps up material into a thin shell with a very strong shock discontinuity in all hydrodynamic variables. These physical discontinuities trigger limiters in the hydrodynamic algorithm, which in turn introduce mathematical discontinuities in the Jacobian of the nonlinear system. When a Lagrangean scheme is utilized to model this problem, the zones which follow the fluid are swept up by the blast wave resulting in few zones in the evacuated region near the origin and many zones near the shock front where physical quantities vary most rapidly. The results of this test are comparable to those of other schemes [25].

5.2. Proto-Neutron Star Cooling Test

In order to demonstrate the applicability of this algorithm to core collapse supernovae and proto-neutron star cooling problems, we have applied the code to a simplified problem that captures the hydrodynamic and radiative behavior of a collapsed stellar core while employing simplified microphysics. The proto-neutron star is constructed to be hot, initially static, and of approximately 1.4 solar masses that is heavily supported by radiation pressure. For the purposes of constructing initial data, the neutrino-radiation distribution is assumed to be static. The multi-group opacity of the material is set to mimic the opacity of that in a more realistic model, while the equation of state is taken to be an ideal gas for the sake of simplicity. The central density of the star is taken to be $\rho_c = 5 \times 10^{14} g/cm^3$. The initial distribution of radiation is chosen so that it contributes approximately one half of the pressure support in the center of the star, while diminishing radially outward from the center. This configuration yields a central temperature of approximately 10 MeV, in agreement with more realistic models of a hot proto-neutron star core taken from supernova studies. The cooling time scale for this problem is set by the opacity and density of the stellar material, which for these problem parameters results in a time scale of several seconds.

We evolve this model in an operator-split fashion through first applying our implicit hydrodynamics algorithm, followed by an implicit multi-group radiation transport algorithm. For comparison, an explicit hydrodynamic CFL restriction would yield a time step of approximately $\Delta t \approx 2.5 \times 10^{-8} s$, hence requiring $O(10^9)$ time steps and rendering even this spherically symmetric problem intractable. In contrast, the implicit algorithm can take a much larger time step since there are no fast-moving shocks or other short time scale phenomena of interest involved in the problem. We utilized the implicit Lagrangean algorithm I to evolve the problem with a CFL factor of 10^3 (i.e. $\Delta t \approx 2.5 \times 10^{-5} s$), with the results shown in Figs. 6-7. After initial transient behavior, the initially static configuration should gradually contract as the neutrino-radiation contribution to the pressure is lost as the neutrinos diffuse outward from the core. This is indeed seen in Figure 6. The neutrino luminosity as a function of time is shown

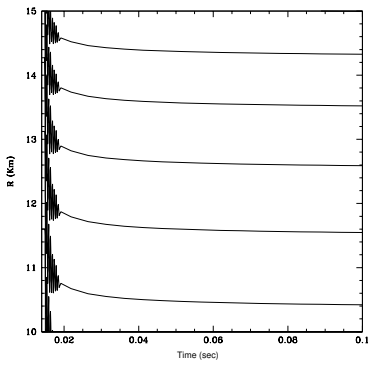

Figure 6. Trajectories of constant-mass contours showing gradual contraction.

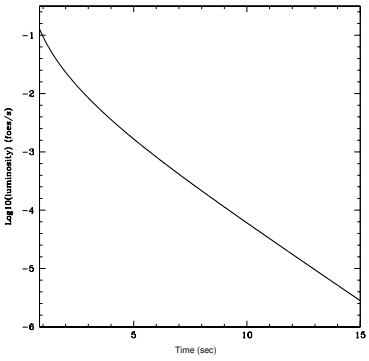

Figure 7. Luminosity of neutrinos in units of foes/s (1 foe = 10^{51} ergs).

in Fig. 7, which correctly decays over the long cooling time scale as the neutrinos gradually diffuse from the core.

6. Conclusions

We have been able to develop a strategy that allows us to utilize NK methods to achieve an implicit solution to a variety of different Lagrangean and Eulerian hydrodynamic simulations. The solution utilizes a finite-difference scheme to acheive a quasi-smooth approximation of the

Jacobian and employs a novel transformation to account for the large dynamic range of physical variables. The use of these methods allows us for the first time to visit multiple timescale problems that could not possibly be attacked via explicit schemes.

Acknowledgements
DRR and CSW gratefully acknowledge support by the Office of Science of the U.S. Dept. of Energy, through the TOPS SciDAC project. FDS gratefully acknowledges support by the Office of Science of the U.S. Dept. of Energy, through SciDAC Award DE-FC02-01ER41185 and supercomputing support for this work at NERSC and NCCS. We would also like to thank Eric Myra for many helpful discussions.

References

[1] Bionta, et al. Observation of a neutrino burst in coincidence with supernova 1987A in the Large Magellanic Cloud. *Phys. Rev. Lett.*, 58:1494–1496, 1987.
[2] K. Hirata, T. Kajita, M. Koshiba, M. Nakahata, and Y. Oyama. Observation of a neutrino burst from the supernova SN1987A. *Phys. Rev. Lett.*, 58:1490–1493, April 1987.
[3] A. Burrows and J. M. Lattimer. The birth of neutron stars. *Astrophys. J*, 307:178–196, August 1986.
[4] F.D. Swesty, J.M. Lattimer, and E.S. Myra. The role of the equation of state in the 'prompt' phase of type II supernovae. *Astrophys. J.*, 425:195–204, 1994.
[5] W. Keil, H.-T. Janka, and E. Mueller. Ledoux Convection in Protoneutron Stars—A Clue to Supernova Nucleosynthesis? *Astrophys. Jl*, 473:L111+, December 1996.
[6] A. Mezzacappa, A. C. Calder, S. W. Bruenn, J. M. Blondin, M. W. Guidry, M. R. Strayer, and A. S. Umar. The Interplay between Proto–Neutron Star Convection and Neutrino Transport in Core-Collapse Supernovae. *Astrophys. J*, 493:848–+, January 1998.
[7] Pons, et al. Evolution of Proto-Neutron Stars. *Astrophys. J*, 513:780–804, March 1999.
[8] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*, volume 16 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, PA, 1995.
[9] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comp. Phys.*, 193:357–397, 2004.
[10] F.D. Swesty and E.S. Myra. A numerical algorithm for modeling multigroup neutrino-radiation hydrodynamics in two spatial dimensions. *Submitted to Astrophys. J.*, 2008.
[11] D. Mihalas and B. Weibel-Mihalas. *Foundations of Radiation Hydrodynamics*. Dover, NY, 1999.
[12] J. M. Stone and M. L. Norman. ZEUS-2D: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions: I. the hydrodynamic algorithms and tests. *Ap. J. Suppl.*, 80:753–790, 1992.
[13] LCPFCT. http://www.lcp.nrl.navy.mil/lcpfct/.
[14] Jay P. Boris and David L. Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *J. Comp. Phys.*, 11:38–69, 1973.
[15] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Statist. Comput.*, 11:450–481, 1990.
[16] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. In A. Iserles, editor, *Acta Numerica*, pages 57–100. Cambridge Univ. Press, New York, 1992.
[17] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986.
[18] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.
[19] P. N. Brown. A local convergence theory for combined inexact-Newton/finite-difference projection methods. *SIAM J. Numer. Anal.*, 24:407–434, 1987.
[20] SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic Solvers). http://www.llnl.gov/casc/sundials.
[21] Hindmarsh, et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, 31(3):363–396, 2005.
[22] Trilinos. http://software.sandia.gov/trilinos.
[23] PETSC (Portable, Extensible Toolkit for Scientific Computation). http://www-unix.mcs.anl.gov/petsc.
[24] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comp. Phys.*, 27:1–31, 1978.
[25] F. D. Swesty. Implicit general relativistic hydrodynamic methods for modeling the late-time explosion mechanism in core collapse supernovae. *Astrophys. J*, 445:811–827, June 1995.